Extracting Temporal Signatures for Comprehending Systems Biology Models

Naren Sundaravaradan[†], K.S.M. Tozammel Hossain[†], Vandana Sreedharan[+], Douglas J. Slotta[†*], John Paul C. Vergara[$], Lenwood S. Heath[†], Naren Ramakrishnan[†]

[†]Department of Computer Science, Virginia Tech, Blackburg, VA 24061, USA
[+]Genetics, Bioinf. and Comp. Biology (GBCB) Program, Virginia Tech, VA 24061, USA
[$]Ateneo de Manila University, Quezon City 1118, Philippines

nares@vt.edu, tozammel@vt.edu, vsree007@vt.edu, dslotta@vt.edu, jpvergara@ateneo.edu, heath@cs.vt.edu, naren@cs.vt.edu

ABSTRACT

Systems biology has made massive strides in recent years, with capabilities to model complex systems including cell division, stress response, energy metabolism, and signaling pathways. Concomitant with their improved modeling capabilities, however, such biochemical network models have also become notoriously complex for humans to comprehend. We propose network comprehension as a key problem for the KDD community, where the goal is to create explainable representations of complex biological networks. We formulate this problem as one of extracting temporal signatures from multi-variate time series data, where the signatures are composed of ordinal comparisons between time series components. We show how such signatures can be inferred by formulating the data mining problem as one of feature selection in rank-order space. We propose five new feaure selection strategies for rank-order space and assess their selective superiorities. Experimental results on budding yeast cell cycle models demonstrate compelling results comparable to human interpretations of the cell cycle.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Time series analysis; H.2.8 [**Database Applications**]: Data mining; I.5.2 [**Design Methodology**]: Feature evaluation and selection

General Terms

Algorithms, Measurement, Experimentation.

*Slotta is now with the National Institute of Mental Health, National Institutes of Health, Bethesda, MD 20892.

Keywords

Temporal signatures, systems biology, feature selection, rank-order spaces, biological networks.

1. INTRODUCTION

Systems biology is an immensely successful enterprise [19] that focuses on modeling biological processes with mathematical formalisms (e.g., ordinary differential equations (ODEs)) and uses numerical simulation and analysis tools to recreate the dynamics of biology. Key processes such as cell division, stress response, energy metabolism, and signaling pathways can now be satisfactorily modeled using systems biology tools. Capabilities such as SBML (Systems Biology Markup Language) and software such as JigCell [24] and COPASI [10] have eased the modeling and simulation process for biologists who might not be trained in the underlying mathematics and algorithmics.

As a case in point, consider the cell division cycle (see Fig. 1 (top left)) of budding yeast (*S. cerevisiae*), which is a model eukaryotic organism. The set of chemical reactions and steps included in the most updated yeast cell cycle mathematical model is highly complex (see Fig. 1 (top left)) involving close to 140 parameters, 48 molecules, and close to 50 equations. The cycle consists of four phases [23]: G1 (Gap 1), S (DNA synthesis), G2 (Gap 2), and M (mitosis). These phases are carefully orchestrated using different protein complexes such as Cdk/cyclin complexes, Cdh1, Sic1, and Cdc6.

Richard Feynman is famously known to have said that "everything that living things do can be understood in terms of the jiggling and wiggling of atoms." Although this can be dismissed as a physicist's reductionist view, the reality is not too far from this quote: the progression through cell cycle phases requires the successive activation and inactivation of protein molecules and complexes. For instance, in the G1 phase, proteins like Cdh1, Sic1, and Cdc6 are more abundant than Cdk/clb complexes, whereas in the S/G2/M phases it is the opposite. Hence, one key way to comprehend the cell cycle is to understand which molecules overpower which others (and when), which molecules are decaying versus which others are increasing in concentration, and which molecules regulate the production/demise of others.

Unfortunately, this type of information is quite difficult to

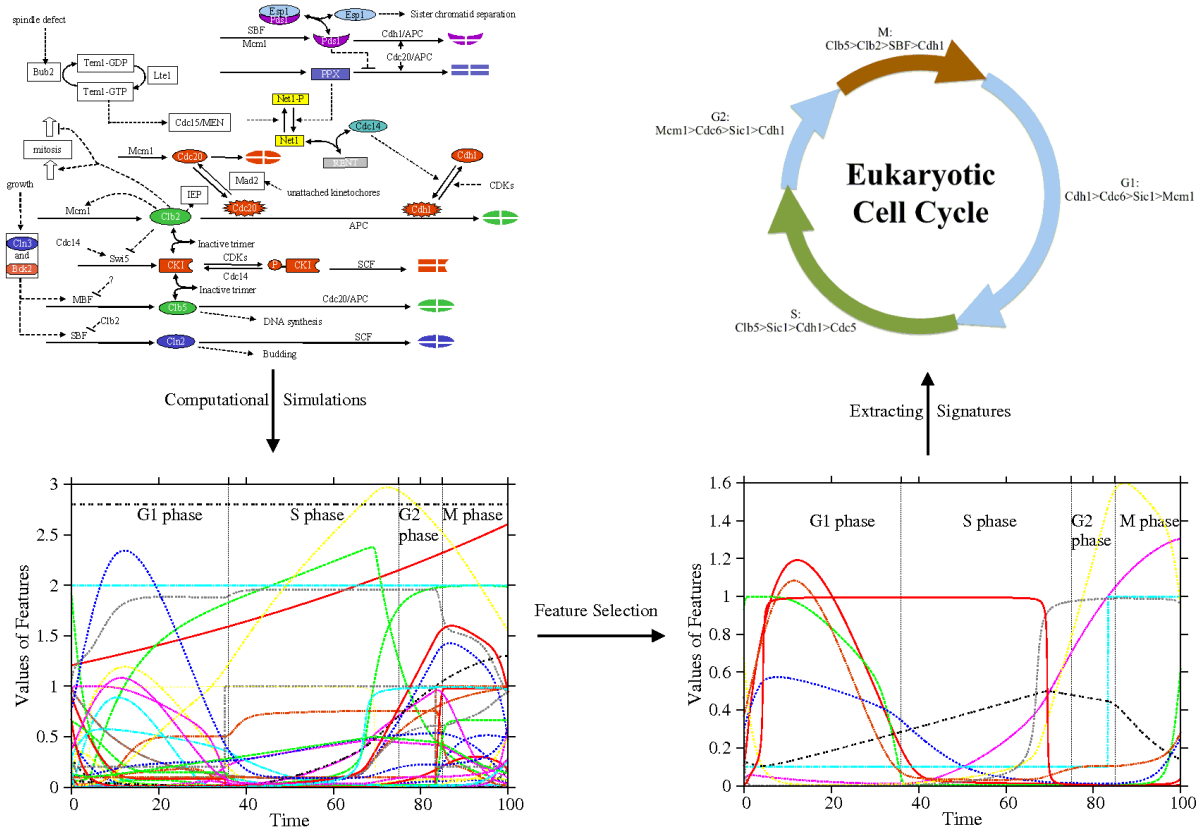

Figure 1: The biological network comprehension problem: a biochemical circuit (top left) is simulated to yield multivariate time course data (bottom left) which is then subjected to feature selection to identify the most significant molecules (bottom right) which are then summarized into temporal signatures of pathway progression (top right). Figure courtesy: (Top left) Chen et al. [5] and (Top right) figure motivated by diagram in [17].

gather from the diagram of Fig. 1 (top left). Because biochemical reaction networks are conceptualizations of dynamical processes, even skilled scientists cannot eyeball these diagrams and comprehend them. One approach is to simulate these networks, generate multi-variate time course data corresponding to the set of molecules, and use data analysis tools to study the time series. This, of course, simply exacerbates the issue, since, without sophisticated data mining tools, comprehending time series of the scale envisaged here is a herculean task.

We hence propose **network comprehension** (Fig. 1) as a key problem for the KDD community: given a biochemical network, mine the multivariate time course data simulated by the network to gain insight into the network's functioning. The key form of insight we seek is understanding of the 'race' conditions between the proteins and how these race conditions drive the internal state of the organism. In particular, which protein molecules contribute to the notion of internal state? What relationships must exist between these molecules in each stage of the dynamical process? We view these relationships as **temporal signatures**, since they serve to summarize the underlying state of the dynamical system.

The problem of extracting temporal signatures can be viewed as feature selection in rank-order space [20]. Here, proteins are the features, instances are vectors of concentration values of proteins, and the classes are the phases of the biological system (e.g., G1, S, G2, M phases). Because the dynamics are driven by relationships between protein concentrations, it is not meaningful to work directly with the amount of proteins in an attribute-value sense. (For instance, in understanding the overall dynamics of the system, it is incomprehensible to work with a measurement quantifying 300-500 molecules of Clb2 as being present in a yeast cell whose volume is just 50e-15 liter.) Instead we rank the features and think of each instance as a total order over the features. The network comprehension problem then reduces to conducting feature selection in rank-order space and constructing temporal signatures by composing ordinal comparisons between protein molecules.

There are many applications that highlight the importance of feature selection in rank-order datasets. In biomedical instrumentation, the desire is to select a subset of electrodes from an EEG dataset and use profiles of relative signal strength as indicators of patient health [26]. Here the instances are the patients, the classes are the diagnoses, and the features denote signal strength as measured using different electrodes. In large-scale gene expression (microarray) assays [2], one possible aim is to classify an experimental condition using expression changes only across a 'salient' subset of genes. For instance, by observing a handful of genes (features) and ranking them by their expression lev-

els, it is possible to qualitatively characterize the cellular transcriptional state (class) for a given condition (instance). In decision-making referendums, one goal is to identify key voting indicators to infer political biases of constituencies. Here, the instances are the constituencies, the classes are political party strongholds, and the features might be socio-economic indicators. On a lighter vein, it appears possible to classify a movie as an art film or a mass market flick by ranking critics! Given such a widespread prevalence of applications where rank order is pertinent [15], it is surprising that this feature selection problem has received little attention so far.

While order-theoretic considerations have been studied in different guises in machine-learning research, our formulation is different from the more traditional settings where instances or classes are ordered or the feature values are ranked across instances [8, 13, 14]. These types of formulations are pertinent in applications such as recommender systems or information retrieval [6, 11], where the goal is to learn and mine a ranking or to infer total orders from given preferences. In our case, features can be ranked within an instance and the goal is to use ordinal comparisons as descriptors for classes.

The key contributions of this paper are:

1. Formulation of the biochemical network comprehension problem as feature selection in rank-order space. Both the application domain and the feature selection problem are novel tasks previously unstudied in the KDD community.

2. Five feature selection strategies—GreedyKL, KS, Spoilers, Center Distance Vector (CDV), and CDV+—that help remove irrelevant and redundant ordinal features from the multivariate time course data. Our strategies are parameter-free and hence do not require arbitrary settings.

3. Experimental results on both synthetic data (to assess the effectiveness of our feature selection strategies) and on real yeast cell cycle data (to illustrate the capabilities of our approach for network comprehension).

2. PRELIMINARIES

Let $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$ be a set of biochemical species (features), and let D be the common domain of S_i. In the case of the yeast cycle, $\mathbf{S}$ includes molecules such as $\{Cln2, Clb2, Clb5, Sic1, Cdc6\}$, and D denotes non-negative reals, to capture molecular concentrations. For simplicity, we assume that a total order is defined on D. A *feature instance* is a tuple $\mathbf{s} = (s_1, s_2, \ldots, s_n) \in D^n$. Any feature instance $\mathbf{s}$ defines an order on $\mathbf{S}$ by the rule $S_i < S_j$ if $s_i < s_j$. Let $C = \{C_1, C_2, \ldots, C_l\}$ be a set of classes. For the yeast cycle, the classes denote the phases, i.e., $C = \{G1, S, G2, M\}$. Then a *dataset* T is a nonempty multiset of pairs $\{(\mathbf{s}, c)\}$, where $\mathbf{s} \in D^n$ and $c \in C$. Suppose, $\mathbf{S}$ consists of the molecules $Cln2, Clb2, Clb5, Sic1$, and $Cdc6$, which have concentrations of 0.07, 0.15, 0.05, 0.02, and 0.11 respectively in $G1$ phase. Then $\mathbf{s} = (0.07, 0.15, 0.05, 0.02, 0.11)$ is a feature instance, and $((0.07, 0.15, 0.05, 0.02, 0.11), G1) \in T$.

2.1 Rank-order space

We recast the given dataset T into a rank-order dataset, whose feature values capture the relative order between pairs of feature values of a feature instance $\mathbf{s}$. We define the *rank-order feature set* for $\mathbf{S}$ to be $\mathbf{F} = \{F_1, F_2, \ldots, F_n\}$, where the domain of each F_i is $\{1, 2, \ldots, n\}$. Let $\pi_{\mathbf{s}} \colon \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$ be a permutation that sorts $\mathbf{s}$ into non-decreasing order. If two features have the same value, we then choose $\pi_{\mathbf{s}}$ arbitrarily from the permutations that satisfy the above condition. Then the *rank-order feature instance* $\mathbf{f}$ for $\mathbf{s}$ has values given by $f_i = \pi_{\mathbf{s}}(i)$. For example, given a feature instance $\mathbf{s} = (0.07, 0.15, 0.05, 0.02, 0.11)$, the corresponding rank-order feature instance is $\mathbf{f} = (3, 5, 2, 1, 4)$.

The *rank-order dataset* is the multiset $T^R = \{(\mathbf{f}, c) \mid (\mathbf{s}, c) \in T\}$. Each dataset T^R implies a probability distribution P as follows:

$$P(\mathbf{f}, c) = \frac{|(\mathbf{f}, c) \in T^R|}{|T^R|}$$

Moreover, if a feature instance $\mathbf{f}$ occurs in at least one pair of T, then the conditional probability

$$P(c \mid \mathbf{f}) = \frac{P(\mathbf{f}, c)}{P(\mathbf{f})}$$

is defined for all $c \in C$. We will use the term $P(C \mid \mathbf{f})$ to refer to the conditional probability distribution of C given the features.

2.2 Irrelevant and redundant features

A classical definition of irrelevant and redundant features is that put forth by Blum and Langley [3]. These notions apply to regular datasets (such as T), rather than rank-order datasets (such as T^R). A feature S_i is *irrelevant* if it has no effect on the class distributions and is independent of other features. A feature S_i is *strongly relevant* if there exists at least two feature instances $\mathbf{s}_1$ and $\mathbf{s}_2$ in T that differ only in their values assigned to S_i and have different classifications, if they appear in T multiple times [3].

These definitions have to be modified for the rank-order case. A feature F_i in rank-order space is a *strongly relevant feature* if there exists at least two data instances $(\mathbf{f}_p, C_m)$, $(\mathbf{f}_q, C_n) \in T^R$ such that when F_i is removed, $\mathbf{f}_p$ and $\mathbf{f}_q$ collapse to the same permutation. For example, the feature F_1 in Fig. 2 is a strongly relevant feature because removing F_1 makes Row 1 and Row 2 of the table to collapse to the same permutation. A feature F_i is a *weakly relevant feature* if there exists a nonempty subset of features $\mathbf{F'} \subset \mathbf{F}$ such that removing $\mathbf{F'}$ makes F_i strongly relevant. For example, the feature F_2 in Fig. 2 is a weakly relevant feature because it becomes strongly relevent if F_1 is removed. We call a feature *redundant* if it is weakly relevant, but not strongly relevant.

S_1	S_2	S_3	S_4	class
20	33	65	40	a
40	25	60	30	b
50	35	20	70	c
80	55	35	40	d

F_1	F_2	F_3	F_4	class
1	2	4	3	a
3	1	4	2	b
3	2	1	4	c
4	3	1	2	d

Figure 2: Dataset T (left table) with its corresponding rank-order dataset T^R (right table).

2.3 Boolean-order space

We define the *boolean-order feature set* for $\mathbf{F}$ to be $\mathbf{B} = \{B_{i,j} \mid 1 \le i < j \le n\}$, where the domain of each $B_{i,j}$ is

$B_{1,2}$	$B_{1,3}$	$B_{1,4}$	$B_{2,3}$	$B_{2,4}$	$B_{3,4}$	class
true	true	true	true	true	false	a
false	true	false	true	true	false	b
false	false	true	false	true	true	c
false	false	false	false	false	true	d

Figure 3: Boolean order dataset T^B corresponding to the rank-order dataset T^R in Fig. 2.

$\{true, false\}$. Given a rank-order feature instance $\mathbf{f}$, the corresponding *boolean-order feature instance* $\mathbf{b}$ has values given by,

$$b_{i,j} = \begin{cases} true & \text{if } f_i < f_j, \\ false & \text{otherwise.} \end{cases}$$

The *boolean-order dataset* for T^R is a multiset $T^B = \{(\mathbf{b}, c) \mid (\mathbf{f}, c) \in T^R\}$. An example of T^B is illustrated in Fig. 3. We will evaluate classification performance of the algorithms using T^B rather than T^R because we need to perform classification on orders and not on the absolute values. While the boolean table might contain redundant features due to transitivity, such transitivity is local to each instance. Given features $B_{1,2}$ ($F_1 < F_2$), $B_{2,3}$ ($F_2 < F_3$), and $B_{1,3}$ ($F_1 < F_3$), if $B_{1,2}$ and $B_{2,3}$ are true in a given instance, then $B_{1,3}$ will be true. But in another instance, $B_{2,3}$ and $B_{1,3}$ can be true without $B_{1,2}$ being true. Hence, $B_{1,3}$ is not redundant here. Thus, it is not possible to universally remove a feature based on transitivity.

2.4 Feature selection criteria

Numerous criteria can be applied for removing redundant and irrelevant features which result in a reduction of $\mathbf{F}$ to a subset $\mathbf{F'} \subset \mathbf{F}$. Such criteria include improved accuracy of predictive modeling, smaller description length for learned mappings, or preservation of as much of the relationship between class distributions and features as possible. We will use the latter criterion in this paper.

For $\mathbf{f} \in T^R$, let $\mathbf{f_{F'}}$ be the projection of $\mathbf{f}$ onto the features in $\mathbf{F'}$. For a dataset T^R, let $T^R_{\mathbf{F'}} = \{\mathbf{f_{F'}} \mid \mathbf{f} \in T^R\}$ be the projection of T^R using the feature set $\mathbf{F'}$. Our goal is to approximate $P(C \mid \mathbf{f})$ with $P(C \mid \mathbf{f_{F'}})$. A popular approach to characterizing the difference between two distribution is the KL-divergence [7]. The KL-divergence between distributions P and Q is $KL(P,Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$, where $0 \log \frac{0}{Q(x)} = 0$ and $P(x) \log \frac{P(x)}{0} = \infty$ whenever $P(x) > 0$. Koller and Sahami defined two feature subset divergence quantities, $\delta_{\mathbf{F'}}$ and $\Delta_{\mathbf{F'}}$ using KL-divergence [12].

For each data instance $\mathbf{f} \in T^R$, divergence is $\delta_{\mathbf{F'}}(\mathbf{f}) = KL(P(C \mid \mathbf{f}), P(C \mid \mathbf{f_{F'}})$. Now, the *feature subset divergence* of $\mathbf{F'}$ is $\Delta_{\mathbf{F'}} = \sum_{\mathbf{f}} P(\mathbf{f}) \delta_{\mathbf{F'}}(\mathbf{f})$, where $P(\mathbf{f})$ is taken from the distribution of data instances in T^R. We can utilize the feature subset divergence in two ways. One way is to define a divergence threshold and search for a smallest subset $\mathbf{F'}$ such that $\Delta_{\mathbf{F'}}$ is at most that threshold. Another is to search for a subset in all $\mathbf{F'} \subset \mathbf{F}$ of a fixed size that minimizes $\Delta_{\mathbf{F'}}$.

In the context of a rank-order dataset, a feature instance $\mathbf{f}$ is a permutation, and we require methods to measure distances between permutations. We adopt two established approaches for defining a distance function between two permutations. The Spearman's distance [21] between two rank-order feature instances $\mathbf{f}_i$ and $\mathbf{f}_j$ is:

$$sd(\mathbf{f}_i, \mathbf{f}_j) = \sum_{k=1}^{n} (\mathbf{f}_i(k) - \mathbf{f}_i(k))^2,$$

where k is the index of the kth feature.

We use Kendall tau distance as our second distance function between two permutations. The Kendall tau distance between two rank-order feature instances $\mathbf{f}_i$ and $\mathbf{f}_j$ is,

$$kd(\mathbf{f}_i, \mathbf{f}_j) = |(x,y) : x < y, (\mathbf{f}_i(x) < \mathbf{f}_i(y) \wedge \mathbf{f}_j(x) > \mathbf{f}_j(y)) \vee (\mathbf{f}_i(x) > \mathbf{f}_i(y) \wedge \mathbf{f}_j(x) < \mathbf{f}_j(y))|$$

Let I be a multiset of rank-order instances that belong to a single class in T^R and let I' be the set I without feature F_i. We define the *spoiler count* $sp(F_i, I)$ for feature F_i with respect to I as follows.

$$sp(F_i, I) = \sum_{\mathbf{f}_i, \mathbf{f}_j \in I'} kd(\mathbf{f}_i, \mathbf{f}_j).$$

Since the instances belong to the same class and order is presumably indicative of class membership, these permutations ought to be similar in order. The feature that has the highest spoiler count contributes most to the differences in order and is therefore a good candidate for removal. For each class $c \in C$, let $T^R(c)$ represent the set of rank-order instances that belong to c in T^R. The *total spoiler count* $tsp(F_i)$ of a rank-order feature F_i is

$$tsp(F_i) = \sum_{c \in C} sp(F_i, T^R(c)).$$

2.5 Centers of permutations

A final notion we will find useful is measures of central tendency around permutations. Let M be a set of permutations. Then the center $ctr(M)$ of M is a permutation π_c (not necessarily in M) that minimizes $\sum_{\pi \in M} sd(\pi, \pi_c)$. Algorithmically, we can compute the permutation $ctr(M)$ by summing the ranks in each position, across all permutations, and deriving $ctr(M)$ from the order of the resulting sums (if there are duplicate sums, ties are broken arbitrarily). Each class C_i for which $M_i = \{\mathbf{f} \mid (\mathbf{f}, C_i) \in T^R\}$ is nonempty, $ctr(M_i)$ is a permutation at the center of a smallest hypersphere containing all the permutations of M_i. If C_i and C_j are distinct classes, then $ctr(P_i)$ and $ctr(P_j)$ are representatives of the two classes that can be used to measure a distance between the two classes. In addition, a center provides an estimate of the expected ranks of each feature in that class, but, more importantly the expected relative orders of the features; a fact which we shall make use of later on. A feature selection algorithm might choose to remove a feature that yields distances that are closest to the original distances.

2.6 Temporal Signatures

Finally, a temporal signature can be viewed as a descriptive summary of the rank-order relationships across classes (typically, such summaries are made after feature selection). There are many ways to define such signatures. In this paper, we take the temporal signature $S_k(c_i)$ of a class label c_i, parameterized by k, as a set $\{T_1, ..., T_k\}$ of k permutation centers derived from all the instances associated with the label. The centers of a class are computed using a k-means algorithm where the distance measure between permutations

S_1	S_2	S_3	class
3.5	4.2	5.7	a
3.5	4.2	1.5	b
3.5	1.5	2.4	c
3.5	2.4	5.7	d

F_1	F_2	F_3	class
1	2	3	a
2	3	1	b
3	1	2	c
2	1	3	d

Figure 4: Datasets T (left table) and T^R (right table) where a removable feature in T does not apply in T^R.

is the Spearman's distance. The signatures thus represent a set of k distinct expected orders capturing the essence of all the instances associated with a class. In the experiments described here, we set $k = 1$; we did not experience significant gain in information with higher values of k.

3. COMPARING FEATURE SPACES

In this section, we investigate relationships among feature spaces, rank-order spaces, and boolean-order spaces. Let S_k be a feature in $\mathbf{S}$, F_k be the corresponding rank-order feature in $\mathbf{F}$, $\mathbf{S'} = \mathbf{S} - \{S_k\}$, and $\mathbf{F'} = \mathbf{F} - \{F_k\}$. We say that F_k is a *removable* feature if $\Delta_{F'} = 0$. We now examine the relationship between $\Delta_{S'}$ and $\Delta_{F'}$. The next two conjectures and their counterexamples illustrate that the relationship is not simple or easily exploited.

CONJECTURE 1. *Let T be a dataset with feature set $\mathbf{S}$ and suppose S_k is a feature such that $\Delta_{\mathbf{S}-\{S_k\}} = 0$. Then, $\Delta_{\mathbf{F}-\{F_k\}} = 0$ in T^R.*

COUNTEREXAMPLE: Figure 4 provides an example of a dataset T and corresponding rank-order dataset T^R. Here, S_1 is a removable feature in dataset T (values assigned to S_1 are the same for all instances). This means $\Delta_{\mathbf{S}-\{S_1\}} = 0$. However, $\Delta_{\mathbf{F}-\{F_1\}} \neq 0$ in T^R, because there is at least one instance $\mathbf{f}$ (such as $\mathbf{f} = (3, 1, 2)$) such that $\delta_{\mathbf{F'}}(\mathbf{f}) \neq 0$.

CONJECTURE 2. *Let T be a dataset with feature set $\mathbf{S}$, and let T^R with feature set $\mathbf{F}$ be its corresponding rank-order dataset. Suppose F_k is a feature such that $\Delta_{\mathbf{F'}} = 0$. Then, $\Delta_{\mathbf{S'}} = 0$ in T.*

COUNTEREXAMPLE: Figure 5 provides an example of a dataset T and corresponding rank-order dataset T^R. Here, F_1 can be removed (as can any single feature) while retaining the same capacity to classify so that $\Delta_{\mathbf{F'}} = 0$ in T^R, where $\mathbf{F'} = \mathbf{F} - \{F_1\}$. However, $\Delta_{\mathbf{S'}} \neq 0$ for $\mathbf{S'} = \mathbf{S} - \{S_1\}$ since S_1 is in fact the feature that distinguishes the two instances in the dataset ($\delta_{\mathbf{S'}}(\mathbf{s}) \neq 0$ for both instances).

S_1	S_2	S_3	class
1	4	5	a
2	4	5	b

F_1	F_2	F_3	class
1	2	3	a
1	2	3	b

Figure 5: Datasets T (left table) and T^R (right table) where a removable feature in T^R does not apply in T.

We now present the following result, which demonstrates that rank-order datasets and boolean-order datasets contain the same order-theoretic information with respect to feature selection.

LEMMA 1. *Let T^R be a rank-order dataset with feature set $\mathbf{F}$, and let T^B with feature set $\mathbf{B}$ be its corresponding boolean order dataset. Furthermore, let $\mathbf{F'} \subset \mathbf{F}$. Define $\mathbf{B'} \subset \mathbf{B}$ to be the set of all features $B_{i,j}$ such that $F_i, F_j \in \mathbf{F'}$. Then, $\Delta_{\mathbf{F'}} = \Delta_{\mathbf{B'}}$.*

PROOF. From the definition of boolean space, it suffices to show that $P(\mathbf{f}) = P(\mathbf{b})$ and $\delta'_{\mathbf{F}}(\mathbf{f}) = \delta'_{\mathbf{B}}(\mathbf{b})$, for all instances $\mathbf{f}$. The equality $P(\mathbf{f}) = P(\mathbf{b})$ follows directly from how the boolean order set was constructed since there is a one-to-one correspondence between rank-order instances ($\mathbf{f}$) and boolean order instances ($\mathbf{b}$). We obtain that

$$\delta'_{\mathbf{F}}(\mathbf{f}) = KL(P(C \mid \mathbf{f}), P(C \mid \mathbf{f}'_{\mathbf{F}})),$$

and

$$\delta'_{\mathbf{B}}(\mathbf{b}) = KL(P(C \mid \mathbf{b}), P(C \mid \mathbf{b}'_{\mathbf{B}}))$$

are equal by the observation that the projections performed on each of the datasets are essentially equivalent. For a given $\mathbf{f}$, $P(C \mid \mathbf{f})$ and $P(C \mid \mathbf{b})$ obviously yield the same distributions, again because of the one-to-one transformation. For the distributions $P(C \mid \mathbf{f}'_{\mathbf{F}})$ and $P(C \mid \mathbf{b}'_{\mathbf{B}})$, on the other hand, we note that a projection in rank-order space preserves the relative order of the features even with the (possible) update in rank values. This in turn corresponds to the boolean order features that are projected in boolean order space. Thus, $P(C \mid \mathbf{f}'_{\mathbf{F}}) = P(C \mid \mathbf{b}'_{\mathbf{B}})$, and the result follows. □

Lemma 1 implies that it is sufficient to consider selection strategies on rank-order datasets and that analogous strategies using boolean order datasets will yield the same results.

4. ALGORITHMS FOR FEATURE SELECTION

We present four feature selection strategies (two taking an information-theoretic approach and the remaining two using discrete mathematics concepts), all of which follow the standard backward stepwise selection framework [9].

$i \leftarrow 0$; $\mathbf{F}_i \leftarrow \mathbf{F}$
while *cond*($\mathbf{F}_i$)
  $F_k \leftarrow h(\mathbf{F}_i)$
  $\mathbf{F}_{i+1} \leftarrow \mathbf{F}_i - \{F_k\}$
  $i \leftarrow i + 1$
end while
return $\mathbf{F}_i$

In this meta-algorithm, the boolean function *cond*($\mathbf{F}_i$) either monitors subset size or subset divergence. The function $h(\mathbf{F}_i)$ is the feature selection function for this selection strategy. This function returns a feature from $\mathbf{F}_i$ in regular feature space, but uses rank order space in its selection process.

Alg 1: (GreedyKL) Use rank-order space, and greedily choose the feature that yields the minimum feature subset divergence when compared against the original dataset. That is, choose $h(\mathbf{F}_i) = F_k$ that minimizes $\Delta_{\mathbf{F}_i - \{F_k\}}$.

Alg 2: (KS) Adapt the Koller-Sahami algorithm [12] for use in rank-order space. First, find (approximate) Markov blankets for all features in the Bayesian network of rank-order features (and class) implied by T^R. A Markov blanket for a set of features $\mathbf{F'}$ is another set of features $\mathbf{G}$

whose values, if known, render $\mathbf{F'}$ independent of all others (i.e., $\mathbf{F} - \mathbf{F'} - \mathbf{G}$) [16]. This term arises from the graphical models literature where a network encodes conditional independencies, and random variables satisfying the above definition form a 'blanket' around the given set of features. In the Koller-Sahami approach, we remove the feature F_k, whose Markov blanket M_k yields the minimum feature subset divergence when compared against $M_k \cup F_k$. That is, $h(\mathbf{F}_i) = F_k$ that minimizes δ_{M_k} with respect to $M_k \cup F_k$.

Alg 3: (Spoilers) A third technique uses the notion of spoilers defined earlier. We use rank-order space and remove the feature with the highest spoiler count. That is $h(\mathbf{F}_i) = F_k$ that maximizes $tsp(F_k)$.

Alg 4: (CDV) The CDV algorithm aims to represent the dataset using the temporal signatures of the classes, and then proceeds to remove features based on their ability to maintain this representation. To construct it we create a set $S = \cup_i S_k(c_i)$ containing the centers of all the classes and then construct a distance vector, D, of all the pairwise distances of the centers in S. D contains $k * \binom{|C|}{2}$ components. Then, for each feature we remove it and recompute this distance vector. In the meta algorithm $h(\mathbf{F}_i) = F_k$, where F_k is the feature that yields the minimum Euclidian distance between the recomputed distance vector and D, the original distance vector.

5. EXPERIMENTAL RESULTS

We now present experimental studies with the above feature selection strategies, including descriptions of datasets, interpretation of results, and discussion.

5.1 Synthetic Datasets

We generate synthetic datasets based on the definitions of irrelevant and redundant features presented earlier. We begin by creating a base set of strongly relevant features and then add a number of redundant and irrelevant features to see if our feature selection algorithms identify and remove them correctly.

Collapsible permutations. Two permutations π_i and π_j are *collapsible* for k if removing $\pi_i(k)$ and $\pi_j(k)$ causes the resulting ranks of the permutations to be equal. For example, let $\pi_i = (4, 1, 3, 2)$ and $\pi_j = (3, 1, 4, 2)$ then p_i and p_j are collapsible for $k = 3$ (i.e., the third element) because the ranks after removal of $p_i(3)$ and $p_j(3)$ yields $(3, 1, 2)$.

To design a feature F_i that is strongly relevant, we simply ensure that removing it will cause the collapse of one or more instances and that the other features are irrelevant when F_i is strong. Each strong feature F_i is designed pertinent to a class distribution made up of two classes c_{i_1} and c_{i_2}. The algorithm to generate the set of base features is:

Algorithm 1 Pseudocode for generating synthetic data.

Input: n: number of strongly relevant features; r: replications
Output: $n * (n + 1)$ instances
  for $i \leftarrow 1$ to n **do**
    $p \leftarrow$ pick a random feature permutation
    $q \leftarrow$ gen permutation to collapse with p for $k = i$
    $ins \leftarrow$ gen n permutations by swapping i with all features in p
    $ins' \leftarrow$ replicate $ins \cup \{p, q\}$ r times
    $insc \leftarrow$ assign classes c_{i_1} or c_{i_2} randomly to ins'

Irrelevant features are incorporated into the base dataset by considering each instance separately and by uniformly inserting the new feature into the existing order implied by the instance.

A redundant feature is incorporated into the dataset by picking a subset of existing strong features F' and using a pure function distorted by normal noise $g(F') + \epsilon$ to generate the ranks for the new feature. In this paper we insert linear features of the form $aF_i + b + \epsilon$ or non-linear features of the form $a \sin(bF_i + c) + \epsilon$. Normal noise is generated with a $\mu = 0$, $\sigma = 1$. For the following experiments, we selected $n = 12$ strong features and $r = 3$ repetitions as input to the base features generation algorithm. We then added irrelevant and redundant features as necessary to this base set of features and instances.

Our first experiment compares the performance of all the algorithms in terms of the number of redundant features they are correctly able to identify. The data consists of 8 redundant features, in which 4 are linear and 4 are non-linear. The linear redundant features are generated using $F_r = aF_i + b + \epsilon$, where a and b are random variables taking values between $(1, 5)$ and ϵ is a normal distribution with $\mu = 0$ and $\sigma = 1$. Non-linear redundant features are generated using $F_r = a \sin(bF_i + c) + \epsilon$ where a, b and c are random variables taking values between $(2, 6)$, $(1, 5)$, and $(1, 5)$ respectively. We generate 100 trials and calculate the percentage of features correctly identified as redundant, F_r (the dependent); and the percentage of features incorrectly identified as redundant, F_i. We see from Fig. 6 (a larger yellow strip is better) that CDV (96%) and Spoilers (87.3%) are the only feature selection strategies that perform well in detecting the actual redundant variables, whereas all the other strategies tend either to detect F_i instead of F_r or none at all. Although, Spoilers performs as well as CDV it is important to note that Spoilers itself is intractable with a larger number of features because of the use of Kendall tau distance to compare two permutations.

Our second experiment compares the performance of all the algorithms in terms of the number of irrelevant features they are correctly able to identify. Here CDV (37.5%) leads in the ability to detect irrelevant features correctly, while all the other algorithms fail to detect even one (see Fig. 6).

5.2 An Improved Algorithm

While CDV performs well on redundant features it does not perform as well in detecting irrelevant features. To get an intuition on why this should be true, consider the nature of the permutation centers and the induced pairwise distance vector. A center of a class is the expected ordering of the features and if a feature has little variance within a class, its removal will have little effect on the resulting center. However, if the feature varies a lot then its removal will completely alter the expected order of the features. This is the case with irrelevant features. Removing them will cause a large change in distances between centers. Since CDV removes features that exhibit the smallest change from the original distance vector, the irrelevant feature continues to remain in the dataset possibly tainting the resulting signatures. This forms the motivation for an improved feature selection approach.

To determine irrelevancy of a feature F_r we adopt a traditional information-theoretic method to rank the features by the reduction in entropy gained from conditioning the data

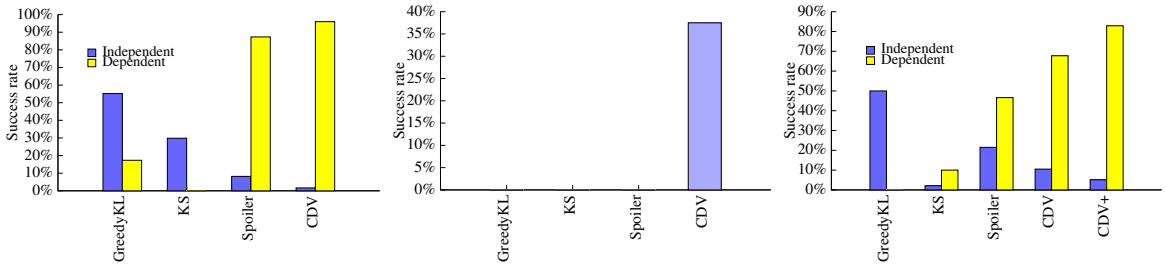

Figure 6: (left) Success rate at removing redundant features in synthetic data. (middle) Success rate at removing irrelevant features in synthetic data.(right)Success rate at removing all features in synthetic data.

on F_r. We calculate this as

$$Z(F_r) = H(C|F_r) + \sum_{k \neq r} H(F_k|F_r),$$

where H is the entropy. The higher the value of $Z(F_r)$ the higher the likelihood that F_r is independent of the features and the class distribution and therefore irrelevant. In the meta-algorithm $h(F_r) = F_k$, where F_k is the feature that yields the maximum Z value.

Alg 5: (CDV+) To mitigate the problems caused by irrelevant features, we first remove features based on Z as defined above, and then perform **CDV** on the rest of the dataset. We refer to this algorithm as **CDV+**.

We now test all the algorithms under the presence of both 4 redundant and 4 irrelevant features on top of the 12 strong features. Over a 100 trials we find that CDV has a positive identification rate of 67.7% while CDV+ has a positive identification rate of 82.9% showing a clear indication that CDV is improved by removing irrelevant features first using a separate information-theoretic algorithm (see Fig. 6). We thus use CDV+ as the feature selection algorithm to process the yeast cell cycle data described next.

5.3 Yeast Cell Cycle

The yeast cell cycle data comes from the mathematical model of the budding yeast cell cycle by Chen et al. [5, 4]. As stated earlier, biochemical interactions are converted to ODEs, which are simulated for different time steps. The concentrations of 45 molecules (recall that these are the features) are recorded for 1000 time steps. In addition to the regular cell cycle data (known as 'normal' or 'wild type'), we also have access to several mutant cell cycles (where one or more key molecules are perturbed and the cell adopts an aberrant state).

The cell cycle is a highly regulated sequence of events: it consists of DNA replication (S phase) and replication of the cell (M phase) separated by two gap phases (G1 and G2). The entire machinery is controlled by four types or categories of molecules [22] as indicated in Fig. 8 (A) (as "SK", "EP", "CDK" and "Enemies").

The entire cell cycle is orchestrated by the coordination among these four kinds of molecules, and that is shown in Fig. 7 (A) where the small squares mark the cell cycle phase. Some of the 45 molecules that we have in our dataset fall under those categories. So our primary task is to see if we can detect the molecules that belong to those four categories. In other words, through our experiments, we seek to determine if conclusions of the form shown in Fig. 8 can be automatically obtained.

Fig. 8(B) shows how the different types of molecules are active at the four phases of the cell cycle. The cell cycle machinery works like a 'see saw' balance between the G1 phase and G2/M/S phases [22]. In other words, most of the molecules that are active in G1 are inactive in the other phases and vice versa. So, the mutual antagonistic nature of the cell cycle phases is expected to be evident from the rank ordering of molecules from our feature selection algorithm.

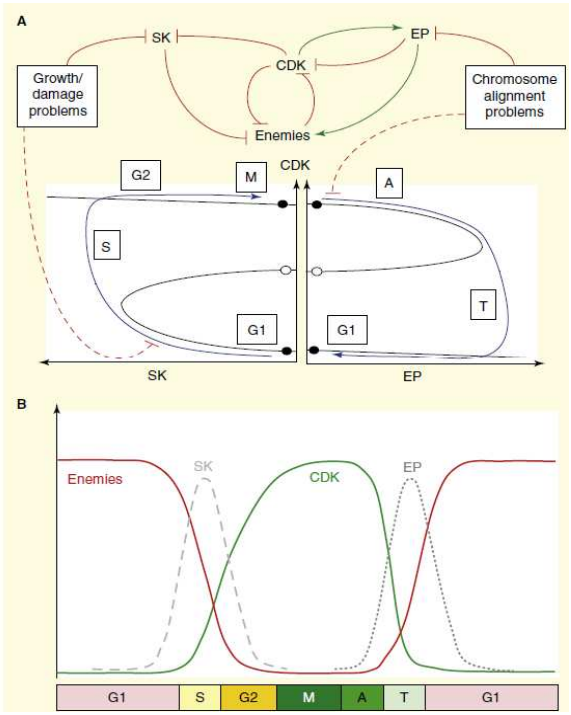

Figure 8: (A) Groups of molecules that control and regulate the cell cycle. (B) Concentrations of these key molecules across all four phases: G1, S, G2 and M. Figure courtesy John Tyson and Bela Novak [22].

In applying feature selection to the cell cycle data, the questions we seek to answer are:

1. Which are the biochemical molecules that play significant roles across all phases of the cell cycle?

2. Is the relative rank order of the reported molecules significant enough to distinguish between the cell cycle phases?

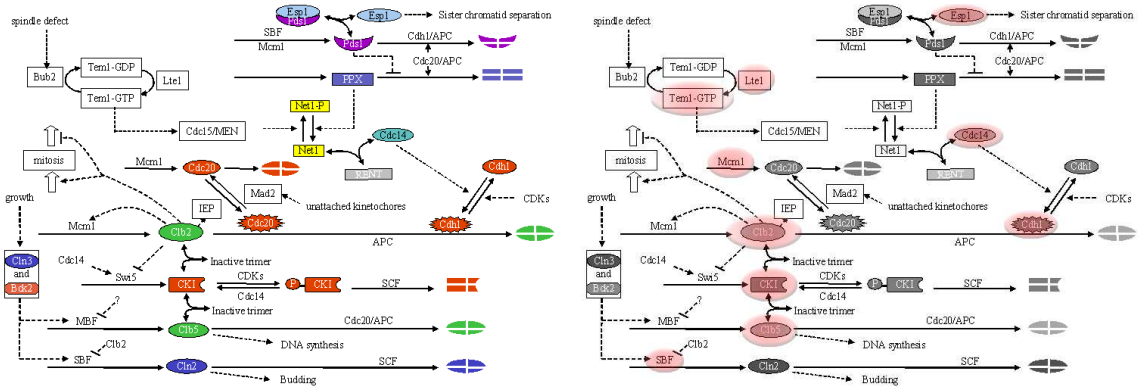

Figure 7: (left) Yeast cell cycle wiring diagram by Chen et al. [5]. (right) The red highlights indicate the significant molecules revealed by the feature selection algorithm as being relevant to distinguishing the cell cycle phases.

Category of molecules in Fig. 8	Molecules which belong to the categories in Fig. 8
CDK	Clb2-Cdc28, Clb5-Cdc28
Enemies	Cdh1, Sic1, Cdc6
SK	Cln1,Cln2,Cln3-Cdc28
EP	Cdc20, Cdc14

Figure 9: Descriptions of molecule categories from Fig. 8.

Category of molecules identified as signatures for each phase	Molecules identified to be significant for each phase of cell cycle
CDK	Clb2T, Clb5T
Enemies	Cdh1, Sic1, Cdc6
SK	SBF (which is a transcription factor for Cln2)
EP	Mcm1 (which is a transcription factor for Cdc20)

Figure 10: Descriptions of molecule categories from Fig. 8 identified by our approach.

3. Can we apply the same analysis to mutant cells (where some of the molecules are knocked out) and can we map out the cell's phenotypic behavior (i.e, observable physical characteristics) by checking the relative rank ordering of the molecules?

From analysis of the wild-type (normal) cell cycle data, CDV+'s results correspond excellently to three of the groups reported in Fig. 8(A). The reported sets of molecules for each of the category in Fig. 8(A) are listed in Fig 9. We report the minimal set of significant molecules that are sufficient to distinguish among the four phases of the cell cycle. Compare Fig. 9 with Fig. 10. One of the molecules missed by our analysis is Cln2, which represents the category "SK". However, this fact was nullified by the presence of another molecule (SBF), which controls Cln2. Comparing Fig. 9 with Fig. 10 reveals that CDV+ successfully identifies all four categories of molecules that the cell cycle machinery is composed of.

Cell cycle mutants

A mutant cell basically means an abnormal cell. Four mutants for the cell cycle were generated by changing parameters in the ODEs corresponding to the molecule being knocked out [5]. These four mutants cause the cell cycle to arrest at different phases.

The collected dataset comprises 5000 instances with 45 feature-values, representing the concentration of each of the molecules at a given time-point. We ran CDV+ with a single centroid in each class. After removing each feature we ran a naive Bayes classifier and tracked its accuracy. Fig. 12 illustrates that the accuracy of the naive Bayes classifier remains almost uniform until the algorithms remove 35 features, which gives us a sense of when to stop removing features. We also note that Fig. 12 shows similar accuracies when running the other algorithms. However, GreedyKL and KS do not remove the correct molecules. On the contrary, the insufficiency of purely information-based removal becomes evident when we inspect the actual molecules removed—Sic1, Cdh1, Mcm1, Cdc20, and Cdc14, many of the key players themselves.

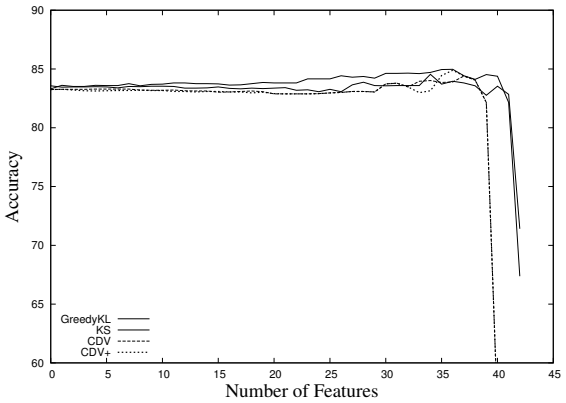

Figure 12: Accuracy of naive Bayes classifier vs feature removed.

Biological interpretations

We validate our results using precise biological facts. Readers who are slighly unfamiliar with the exact molecule names that are mentioned here can picture them as placeholders from Figure 10 and focus on just the interconnection between them resulting in the observed behavior. Names

Phase	Relative rank ordering
Normal cell	
G1	Cdc6 > Cdh1 > SBF > Sic1T > ESP1 > Clb5T > LTE1 > SPN > Clb2T > Mcm1
S	SBF > Clb5T > LTE1 > ESP1 > SPN > Clb2T > Sic1T > Mcm1 > Cdc6 > Cdh1
G2	Mcm1 > Clb2T > SPN > Clb5T > LTE1 > Sic1T > SBF > ESP1 > Cdc6 > Cdh1
M	Clb2T > SPN > Mcm1 > LTE1 > Clb5T > Sic1T > ESP1 > Cdh1 > SBF > Cdc6
Cdc20 mutant	
G1	Cdh1 > Sic1 > C5 > TEM1 > Cdc6P > SPN >PDS1> Clb2T > VI20 > Clb5
S/G2/arrest phase	VI20 > CLB5 > SPN > PDS1 > Cdc6P > Clb2T > TEM1 > C5 > CDH1 > Sic1
Cdc14 mutant	
G1	Cdh1 > RENT > Cdc15 > C5 > LTE1 > Cdc6P > SPN > PDS1 > Clb2T > VI20
S/G2/arrest phase	VI20 > PDS1 > LTE1 > SPN > Cdc6P > Clb2T > C15 > RENT > C5 > Cdh1
Tem1 mutant	
G1	Cdc6 > Cdh1 > Sic1T > ESP1 > Swi5 > PPX > Clb5T > SPN > PDS1 > VI20
S/G2/arrest phase	VI20 > Clb5T > PPX > PDS1 > Swi5 > ESP1 > SPN > Sic1T > Cdc6 > Cdh1
Clb1 and Clb2 mutant	
G1	Cdc6 > Cdh1 > Sic1T > ESP1 > Swi5 > PPX > BUB2 > Clb5T > ORI > VI20
S/G2/arrest phase	VI20 > ORI > BUB2 > CLB5T > PPX > ESP1 > Swi5 > Sic1T > Cdc6 > Cdh1

Figure 11: Signatures extracted by the CDV+ algorithm.

of molecules are written exactly as they appear in Fig. 8 through Fig 11.

- In a normal cell, the G1 phase is biologically distinguishable with the abundance of molecules Cdh1 and Sic1 and Cdc6 [5]. This is shown by high relative rank orders for these molecules in the G1 phase (Normal cells G1 row in Fig 11) [22]. These molecules cause suppression of Clb2. Clb2 is suppressed also because Mcm1 is inactive. Both these factors are shown in G1 phase with low ranks for Mcm1 and Clb2.

  In S phase (Normal cells S row in Fig. 11), Clb5 starts to accumulate, the concentration of Cdh1 starts going down, and as a result Clb2 starts to accumulate in late S phase, and also Sic1 concentration starts decreasing. These factors are sufficient to distinguish the S phase [22]. The S phase is marked by Clb5 concentration going higher, and Mcm1, Cdc6, and Cdh1 having relatively lower rank order.

  In G2 phase (Normal cells G2 in Fig. 11), Mcm1 is of higher significance because it synthesizes another component Clb2 [1]. Mcm1 and Cdh1 share opposite trends. This is evident from the higher order rank for Mcm1 than Cdh1 in G2. This trend is opposite to the trend visible in G1.

  The M phase is characterized by accumulated Clb2, Clb5, and Mcm1, much higher than SBF, Sic1, Cdh1 and Cdc6 [22]. This supports the relative rank ordering found by the algorithm (Normal cells M row in Fig. 11).

- The Cdc20 mutant cells have a G1 phase comparable to normal cells according to the features selected. The S/G2/arrest phase characterization indicates that Cdc20 is required for the degradation of PDS1. In the absence of Cdc20, PDS1 exists at a higher concentration than normal [25]. Therefore, PDS1 is a distinguishing feature of Cdc20 mutants. The results from the feature selection algorithm reveal precisely this, because PDS1 is given a higher rank in the Cdc20 mutant cells but not in the normal cells. Therefore, our algorithm reports that PDS1 is a feature of Cdc20 mutants differentiating it from normal cells, which agrees with the literature. Cdc20 also causes degradation of Clb2 and Cl5. Thus, Clb2 and Clb5 get reported as significant molecules after the cells gets into the arresting phase when there is no Cdc20 in the cell.

- The Cdc14 mutant cells also show that the G1 phase has features comparable to normal cells. S/G2/arrest phase shows that Net1 and Cdc14 are involved in formation of the complex called RENT. Therefore, in the absence of Cdc14 mutant, concentration of RENT differs from the normal cell. Cdc15 is both an activator and substrate for Cdc14. So in the absence of Cdc14, concentration of Cdc15 is different from that in normal cells.

- The Tem1 mutant cells are distinguishable from normal cells due to high PDS1 and ESP1. Concentration of Cdc15 becomes lower than in a normal cell and so concentration of RENT becomes high. Cdh1 is lower and as a result, PDS1 becomes higher in Tem1 than in normal cells. Since Clb5 and Clb2 become higher, Sic1 will be low and becomes another distinguishing signature of the mutant from the normal cell.

- The Clb1 and Clb2 double mutant (represented as Clb2 mutant in the model) does not go into G2 phase, because the deletion causes higher Clb5 and PDS1 concentrations than in normal cells. This also will be caused and indicated by lower ESP1, lower Sic1, and lower Cdh1 than those in normal cells at the corresponding time points.

6. DISCUSSION

We have introduced the novel KDD problem of network comprehension and cast it as feature selection in rank-order space followed by summarizing the remaining features into temporal signatures. We have presented five specific feature selection algorithms for removing redundant and irrelevant

features in rank-order data: our results demonstrate that rank-order data are better analyzed using an order-theoretic algorithm (CDV+) versus traditional information-theoretic algorithms (e.g., KS). Both our synthetic and real-world studies reveal that CDV+ is the best performing algorithm in terms of picking out the redundant and irrelevant features, and not just maintaining a high classification accuracy rate.

Our future work is focused on delving further into network comprehension goals. While temporal signatures yield a great deal of insight into the functioning of cellular biology, taking into account the regulatory mechanisms, for example, can help us assess if the rank order changes from one phase to another are triggered by or correlate with changes in some regulatory molecules. Ultimately, just like an electrical engineer uses metaphors of amplifiers, oscillators, and switches to comprehend an unknown circuit, we wish to design decompositions of biochemical circuits as compositions of input-output signals mediated by protein molecules. The initial steps toward these ideas have been taken [18, 19] and this area is ripe for the development and application of data mining methods.

Acknowledgments

This work is supported in part by US NSF grants CCF-0937133, CNS-0615181, ITR-0428344, and the Institute for Critical Technology and Applied Science (ICTAS), Virginia Tech.

7. REFERENCES

[1] A. Amon, M. Tyers, B. Futcher, and K. Nasmyth. Mechanisms that help the yeast cell cycle clock tick: G2 cyclins transcriptionally activate G2 cyclins and repress G1 cyclins. *Cell*, 74(6):993–1007, 1993.

[2] M.N. Arbeitman et al. Gene expression during the life cycle of *Drosophila melanogaster*. *Science*, 297(5590):2270–2275, September 2002.

[3] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[4] K.C. Chen, A. Csikasz-Nagy, B. Gyorffy, J. Val, B. Novak, and J.J. Tyson. Kinetic analysis of a molecular model of the budding yeast cell cycle. *Molecular Biology of the Cell*, 11(1):369–391, 2000.

[5] K.C. Chen et al. Integrative analysis of cell cycle control in budding yeast. *Molecular Biology of the Cell*, 15(8):3841–3862, 2004.

[6] W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *JAIR*, 10:243–270, 1999.

[7] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, August 1991.

[8] X. Geng, T. Liu, T. Qin, and H. Li. Feature selection for ranking. In *Proc. SIGIR'07*, pages 407–414, 2007.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Satistical Learning*. Springer-Verlag, 2001.

[10] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI–a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[11] T. Kamishima. Nantonac collaborative filtering: Recommendation based on order responses. In *Proc. KDD'03*, pages 583–588. ACM Press, 2003.

[12] D. Koller and M. Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.

[13] G. Lebanon and J. Lafferty. Cranking: Combining rankings using conditional probability models on permutations. In *International Conference on Machine Learning*, pages 363–370, 2002.

[14] G. Lebanon and J. Lafferty. Conditional models on the ranking poset. In *Advances in Neural Information Processing Systems 15*, pages 431–438, 2003.

[15] J.I. Marden. *Analyzing and Modeling Rank Data*. CRC Press, Aug 1996.

[16] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, September 1997.

[17] Orians G.H. Heller H.C. Purves, W. K. *Life: The Science of Biology*. W H Freeman & Co, 4th edition, 1994.

[18] N. Ramakrishnan and U.S. Bhalla. Memory switches in chemical reaction space. *PLoS Computational Biology*, 4(7), 2008.

[19] N. Ramakrishnan, U.S. Bhalla, and J.J. Tyson. Computing with proteins. *Computer*, 42(1):47–56, 2009.

[20] D.J. Slotta, J.P. Vergara, N. Ramakrishnan, and L.S. Heath. Algorithms for feature selection in Rank-order spaces. Technical report, Department of CS, VT, 2005.

[21] C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15:72–101, 1904.

[22] J.J. Tyson and B. Novak. Temporal organization of the cell cycle. *Current Biology*, 18(17):759–768, 2008.

[23] J.J. Tyson, B. Novak, K.C. Chen, and J. Val. Checkpoints in the cell cycle from a modeler's perspective. *Progress in Cell Cycle Research*, 1:1–8, 1995.

[24] M. Vass, N. Allen, C.A. Shaffer, N. Ramakrishnan, L.T. Watson, and J.J. Tyson. The JigCell model builder and run manager. *Bioinformatics*, 20(18):3680–3681, 2004.

[25] R. Visintin, S. Prinz, and A. Amon. CDC20 and CDH1: A family of substrate-specific activators of APC-dependent proteolysis. *Science*, 278(5337):460, 1997.

[26] X.L. Zhang, H. Begleiter, B. Porjesz, W. Wang, and A. Litke. Event related potentials during object recognition tasks. *Brain Research Bulletin*, 38(6):531–538, 1995.